C

Closed Itemset Mining and Non-redundant Association Rule Mining

Mohammed J. Zaki
Rensselaer Polytechnic Institute, Troy, NY, USA

Synonyms

Frequent concepts; Rule bases

Definition

Let I be a set of binary-valued attributes, called *items*. A set $X \subseteq I$ is called an *itemset*. A transaction database D is a multiset of itemsets, where each itemset, called a transaction, has a unique identifier, called a tid. The *support* of an itemset X in a dataset D, denoted $sup(X)$, is the fraction of transactions in D where X appears as a subset. X is said to be a *frequent* itemset in D if $sup(X) \geq minsup$, where *minsup* is a user defined minimum support threshold. An (frequent) itemset is called *closed* if it has no (frequent) superset having the same support.

An *association rule* is an expression $A \Rightarrow B$, where A and B are itemsets, and $A \cap B = \varnothing$. The *support* of the rule is the joint probability of a transaction containing both A and B, given as $sup(A \Rightarrow B) = P(A \wedge B) = sup(A \cup B)$. The *confidence* of a rule is the conditional probability that a transaction contains B, given that it contains A, given as: $conf\,(A \Rightarrow B) = P\,(B|A) = \frac{P(A \wedge B)}{P(A)} = \frac{\sup(A \cup B)}{\sup(A)}$. A rule is frequent if the itemset $A \cup B$ is frequent. A rule is confident if $conf \geq minconf$, where *minconf* is a user-specified minimum threshold. The aim of non-redundant association rule mining is to generate a *rule basis*, a small, non-redundant set of rules, from which all other association rules can be derived.

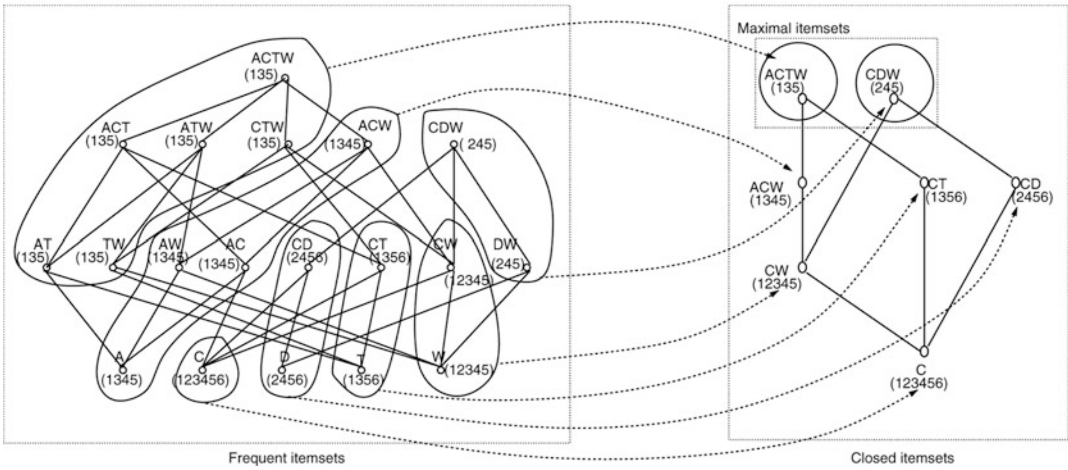
Historical Background

The notion of closed itemsets has its origins in the elegant mathematical framework of Formal Concept Analysis (FCA) [3], where they are called *concepts*. The task of mining frequent closed itemsets was independently proposed in [7, 11]. Approaches for non-redundant association rule mining were also independently proposed in [1, 9]. These approaches rely heavily on the seminal work on rule bases in [5, 6]. Efficient algorithms for mining frequent closed itemsets include CHARM [10], CLOSET [8] and several new approaches described in the Frequent Itemset Mining Implementations workshops [4].

Foundations

Let $I = \{i_1, i_2, \ldots, i_m\}$ be the set of items, and let $T = \{t_1, t_2, \ldots, t_n\}$ be the set of tids, the

L. Liu, M.T. Özsu (eds.), *Encyclopedia of Database Systems*,
DOI 10.1007/978-1-4899-7993-3_66-2



Closed Itemset Mining and Non-redundant Association Rule Mining, Fig. 1 Frequent, closed frequent and maximal frequent itemsets

transaction identifiers. Just as a subset of items is called an itemset, a subset of tids is called a tidset. Let $\mathbf{t} : 2^I \rightarrow 2^T$ be a function, defined as follows:

$$\mathbf{t}(X) = \{t \in T | X \subseteq \mathbf{i}(t)\}$$

That is, $\mathbf{t}(X)$ is the set of transactions that contain *all* the items in the itemset X. Let $\mathbf{i} : 2^T \rightarrow 2^I$ be a function, defined as follows:

$$\mathbf{t}(Y) = \{i \in I | \forall t \in Y, t \text{ contains } x\}$$

That is, $\mathbf{i}(T)$ is the set of items that are contained in *all* the tids in the tidset Y. Formally, an itemset X is closed if $\mathbf{i} \circ \mathbf{t}(X) = X$, i.e., if X is a fixed-point of the closure operator $\mathbf{c} = \mathbf{i} \circ \mathbf{t}$. From the properties of the closure operator, one can derive that X is the maximal itemset that is contained in all the transactions $\mathbf{t}(X)$, which gives the simple definition of a closed itemset, namely, a closed itemset is one that has no superset that has the same support.

Based on the discussion above, three main families of itemsets can be distinguished. Let $\mathcal{F}$ denote the set of all frequent itemsets, given as

$$\mathcal{F} = \{X | X \subseteq I \text{ and } sup(\mathrm{X}) \geq minsup\}$$

Closed Itemset Mining and Non-redundant Association Rule Mining, Table 1 Example transaction dataset

	i(t)
1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT

Let $\mathcal{C}$ denote the set of all closed frequent itemsets, given as

$$\mathcal{C} = \{X | X \in \mathcal{F} \text{ and } \nexists \mathrm{Y} \supset \mathrm{X} \text{ with } sup(\mathrm{X}) = sup(\mathrm{Y})\}$$

Finally, let $\mathcal{M}$ denote the set of all *maximal* frequent itemsets, given as

$$\mathcal{M} = \{X | X \in \mathcal{F} \text{ and } \nexists \mathrm{Y} \supset \mathrm{X} \text{ such that } \mathrm{Y} \in \mathcal{F}\}$$

The following relationship holds between these sets: $\mathcal{M} \subseteq \mathcal{C} \subseteq \mathcal{F}$, which is illustrated in Fig. 1, based on the example dataset shown in Table 1 and using minimum support $minsup = 3$. The *equivalence classes* of itemsets that have the same tidsets have been shown clearly; the largest itemset in each equivalence class is a closed itemset. The figure also shows that the maximal itemsets are a subset of the closed itemsets.

Mining Closed Frequent Itemsets

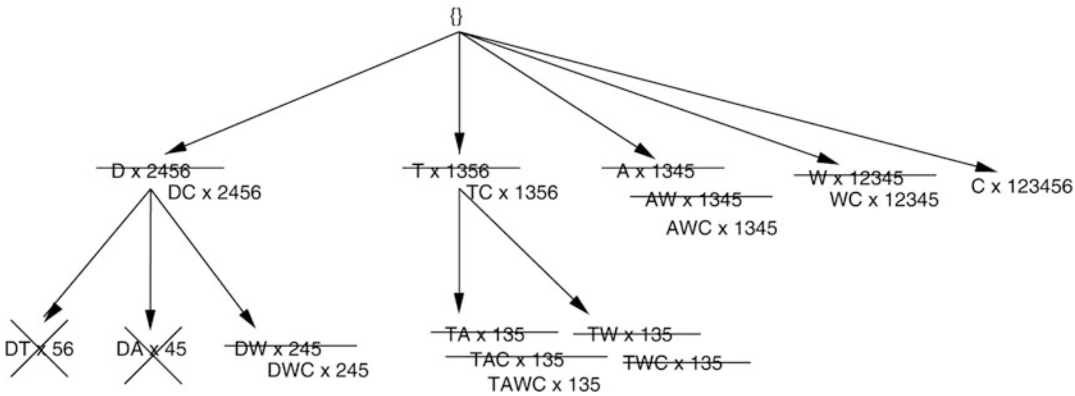
CHARM[8] is an efficient algorithm for mining closed itemsets. Define two itemsets X,Y of length k as belonging to the same *prefix equivalence class*, $[P]$, if they share the $k-1$ length prefix P, i.e., $X = Px$ and $Y = Py$, where $x,y \in I$. More formally, $[P] = \{Px_i \mid x_i \in I\}$, is the class of all itemsets sharing P as a common prefix. In CHARM there is no distinct candidate generation and support counting phase. Rather, counting is simultaneous with candidate generation. For a given prefix class, one performs intersections of the tidsets ofall pairs of itemsets in the class, and checks if the resulting tidsets have cardinality at least *minsup*. Each resulting frequent itemset generates a new class which will be expanded in the next step. That is, for a given class of itemsets with prefix P, $[P] = \{Px_1, Px_2, \ldots, Px_n\}$, one performs the intersection of Px_i with all Px_j with $j > i$ to obtain a new class $[Px_i] = [P']$ with elements $P'x_j$ provided the itemset $Px_i\, x_j$ is frequent. The computation progresses recursively until no more frequent itemsets are produced. The initial invocation is with the class of frequent single items (the class $[\varnothing]$). All tidset intersections for pairs of class elements are computed. However in addition to checking for frequency, CHARM eliminates branches that cannot lead to closed sets, and grows closed itemsets using subset relationships among tidsets. There are four cases: if $\mathbf{t}(X_i) \subset \mathbf{t}(X_j)$ or if $\mathbf{t}(X_i) = \mathbf{t}(X_j)$, then replace every occurrence of X_i with $X_i \cup X_j$, since whenever X_i occurs X_j also occurs, which implies that $\mathbf{c}(X_i) \subseteq \mathbf{c}(X_i \cup X_j)$. If $\mathbf{t}(X_i) \supset \mathbf{t}(X_j)$ then replace X_j for the same reason. Finally, further recursion is required if $\mathbf{t}(X_i) \neq \mathbf{t}(X_j)$. These four properties allow CHARM to efficiently prune the search tree (for additional details see [10]).

Figure 2 shows how CHARM works on the example database shown in Table 1. First, CHARM sorts the items in increasing order of support, and initializes the root class as $[\varnothing] = \{D \times 2456, T \times 1356, A \times 1345, W \times 12345, C \times 123456\}$. The notation $D \times 2456$ stands for the itemset D and its tidset $\mathbf{t}(D) = \{2,4,5,6\}$. CHARM first processes the node $D \times 2456$; it will be combined with the sibling elements. DT and DA are not frequent and are thus pruned. Looking at W, since $\mathbf{t}(D) \neq \mathbf{t}(W)$, W is inserted in the new equivalence class $[D]$. For C, since $\mathbf{t}(D) \subset \mathbf{t}(C)$, all occurrences of D are replaced with DC, which means that $[D]$ is also changed to $[DC]$, and the element DW to DWC. A recursive call with class $[DC]$ is then made and since there is only a single itemset DWC, it is added to the set of closed itemsets $\mathcal{C}$. When the call returns to D (i.e., DC) all elements in the class have been processed, so DC itself is added to $\mathcal{C}$.

When processing T, $\mathbf{t}(T) \neq \mathbf{t}(A)$, and thus CHARM inserts A in the new class $[T]$. Next it finds that $\mathbf{t}(T) \neq \mathbf{t}(W)$ and updates $[T] = \{A,W\}$. When it finds $\mathbf{t}(T) \subset \mathbf{t}(C)$ it updates all occurrences of T with TC. The class $[T]$ becomes $[TC] = \{A,W\}$. CHARM then makes a recursive call to process $[TC]$. When combining TAC with TWC it finds $\mathbf{t}(TAC) = \mathbf{t}(TWC)$, and thus replaces TAC with $TACW$, deleting TWC at the same time. Since $TACW$ cannot be extended further, it is inserted in$\mathcal{C}$. Finally, when it is done processing the branch TC, it too is added to $\mathcal{C}$. Since $\mathbf{t}(A) \subset \mathbf{t}(W) \subset \mathbf{t}(C)$ no new recursion is made; the final set of closed itemsets $\mathcal{C}$ consists of the uncrossed itemsets shown in Fig. 2.

Non-redundant Association Rules

Given the set of closed frequent itemsets $\mathcal{C}$, one can generate all non-redundant association rules. There are two main classes of rules: (i) those that have 100 % confidence, and (ii) those that have less than 100 % confidence [9]. Let X_1 and X_2 be closed frequent itemsets. The 100 % confidence rules are equivalent to those directed from X_1 to X_2, where $X_2 \subseteq X_1$, i.e., from a superset to a subset (not necessarily proper subset). For example, the rule $C \Rightarrow W$ is equivalent to the rule between the closed itemsets $\mathbf{c}(W) \Rightarrow \mathbf{c}(C) \equiv CW \Rightarrow C$. Its support is $sup(CW) = 5/6$, and its confidence is $\frac{\sup(CW)}{\sup(W)} = 5/5 = 1$, i.e., 100 %. The less than 100 % confidence rules are equivalent to those from X_1 to X_2 where $X_1 \subset X_2$, i.e., from a subset to a proper superset. For example, the rule $W \Rightarrow T$ is equivalent to the rule $\mathbf{c}(W) \Rightarrow \mathbf{c}(W \cup T) \equiv CW \Rightarrow ACTW$. Its support is $sup(TW) = 3/6 = 0.5$, and its confidence is $\frac{\sup(TW)}{\sup(W)} = 3/5 = 0.6$ or 60 %. More details on how to generate these non-redundant rules appears in [9].



Closed Itemset Mining and Non-redundant Association Rule Mining, Fig. 2 CHARM: mining closed frequent itemsets

Key Applications

Closed itemsets provide a loss-less representation of the set of all frequent itemsets; they allow one to determine not only the frequent sets but also their exact support. At the same time they can be orders of magnitude fewer. Likewise, the non-redundant rules provide a much smaller, and manageable, set of rules, from which all other rules can be derived. There are numerous applications of these methods, such as market basket analysis, web usage mining, gene expression pattern mining, and so on.

Future Directions

Closed itemset mining has inspired a lot of subsequent research in mining compressed representations or summaries of the set of frequent patterns; see [2] for a survey of these approaches. Mining compressed pattern bases remains an active area of study.

Experimental Results

A number of algorithms have been proposed to mine frequent closed itemsets, and to extract non-redundant rule bases. The Frequent Itemset Mining Implementations (FIMI) Repository contains links to many of the latest implementations for mining closed itemsets. A report on the comparison of these methods also appears in [4]. Other implementations can be obtained from individual author's websites.

Data Sets

The FIMI repository has a number of real and synthetic datasets used in various studies on closed itemset mining.

Url to Code

The main FIMI website is at http://fimi.cs.helsinki.fi/, which is also mirrored at: http://www.cs.rpi.edu/~zaki/FIMI/

Cross-References

- ▶ Association Rule Mining on Streams
- ▶ Data Mining

Recommended Reading

1. Bastide Y, Pasquier N, Taouil R, Stumme G, Lakhal L. Mining minimal non-redundant association rules using frequent closed itemsets. In: Proceedings of the 1st International Conference Computational Logic; 2000. p. 972–86.

2. Calders T, Rigotti C, Boulicaut J-F. A survey on condensed representation for frequent sets. In: Boulicaut J-F, De Raedt L, Mannila H, editors. Constraint-based mining and inductive databases, LNCS, vol. 3848. Berlin: Springer; 2005. p. 64–80.
3. Ganter B, Wille R. Formal concept analysis: mathematical foundations. Berlin/Heidelberg/New York: Springer; 1999.
4. Goethals B, Zaki MJ. Advances in frequent itemset mining implementations: report on FIMI'03. SIGKDD Explor. 2003;6(1):109–17.
5. Guigues JL, Duquenne V. Familles minimales d'implications informatives resultant d'un tableau de donnees binaires. Math Sci Hum. 1986;24(95):5–18.
6. Luxenburger M. Implications partielles dans un contexte. Math Inf Sci Hum. 1991;29(113):35–55.
7. Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed itemsets for association rules. In: Proceedings of the 7th International Conference on Database Theory; 1999. p. 398–416.
8. Pei J, Han J, Mao R. Closet: an efficient algorithm for mining frequent closed itemsets. In: Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery; 2000. p. 21–30.
9. Zaki MJ. Generating non-redundant association rules. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2000. p. 34–43.
10. Zaki MJ, Hsiao CJ. CHARM: an efficient algorithm for closed itemset mining. In: Proceedings of the SIAM International Conference on Data Mining; 2002. p. 457–73.
11. Zaki MJ, Ogihara M. Theoretical foundations of association rules. In: Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery; 1998.